AMENDMENT TO THE CLAIMS

1. (Currently Amended) A re-scheduling system comprising:

a re-scheduler device coupled to an instruction queue to receive an instruction;

a delay unit coupled to said ~~re-scheduling multiplexer~~ re-scheduler device to store a wait history for said instruction; and

a delay queue coupled to ~~the~~ an output of said re-scheduler device to hold said instruction prior to writing said instruction into a scheduler.

2. (Original) The system of claim 1 wherein said delay queue comprises:

said at least one delay line, wherein each of said at least one delay line is to hold said instruction for a fixed number of clock cycles.

3. (Original) The system of claim 2 wherein said re-scheduler device is to sort said instruction into at least one delay line based on said wait history for said instruction.

4. (Original) The system of claim 1 wherein said delay unit is to store wait history information for said instruction when said instruction is first scheduled, and is to update wait history information for said instruction when said instruction is executed.

5. (Original) The system of claim 1 wherein said delay unit is to store latency information for said instruction when said instruction is first scheduled, and is to update latency information for said instruction when said instruction is executed.

6. (Original) The system of claim 1 wherein said delay unit is to store resource conflicts for said instruction when said instruction encounters a resource conflict during execution.

7. (Original) The system of claim 2 wherein said delay block utilizes a general prediction scheme to determine a number of clock cycles to store as said wait history.

8. (Original) The system of claim 7 wherein said re-scheduler device sorts said instruction based on said wait history for said instruction.

9. (Original) A computer processing system comprising:

a memory device to store instructions;

a processor coupled to said memory device and to execute said instructions, said processor including:

a re-scheduler having a first input coupled to an instruction queue, a second input coupled to a replay unit, and an output;

a scheduler coupled to said re-scheduler output;

a replay system having a first and second outputs, said first output coupled to an execution unit and second output coupled to said replay unit;

wherein said re-scheduler comprises:

a re-scheduler device coupled to an instruction queue to receive an instruction;

a delay unit coupled to said re-scheduler device to store wait history for said instruction; and

a delay queue coupled to the output of said re-scheduler device to hold said instruction for a fixed number of clock cycles.

10. (Original) The system of claim 9 wherein said delay queue comprises:
said at least one delay line, wherein each of said at least one delay line holds said instruction for a fixed number of clock cycles.

11. (Original) The system of claim 10 wherein said re-scheduler device is to sort said instruction into at least one delay line based on said wait history for said instruction.

12. (Original) The system of claim 9 wherein said delay unit is to store wait history information for said instruction when said instruction is first scheduled, and is to update wait history information for said instruction when said instruction is executed.

13. (Original) The system of claim 9 wherein said delay unit is to store latency information for said instruction when said instruction is first scheduled, and is to update latency information for said instruction when said instruction is executed.

14. (Original) The system of claim 9 wherein said delay unit is to store resource conflicts for said instruction when said instruction encounters a resource conflict during execution.

15. (Original) The system of claim 10 wherein said delay unit utilizes a general prediction scheme to determine a number of clock cycles to store as said wait history.

16. (Original) The system of claim 15 wherein said re-scheduler device is to sort said instruction based on said wait history for said instruction.

17. (Original) A method of processing a computer instruction in a replay loop comprising:
receiving an instruction output from an instruction queue;
checking a delay unit for a scheduling history for said instruction;
determining a wait time for said instruction;
placing said instruction in a delay queue, wherein said delay queue includes at least one delay line with a fixed wait time; and
writing said instruction to said scheduler after said fixed wait time.

18. (Original) The method of claim 17 wherein determining a wait time for said instruction is based on said scheduling history stored in said delay unit.

19. (Original) The method of claim 18 wherein placing said instruction in a delay queue comprises:
matching said wait time for said instruction to the closest said fixed wait time for said at least one delay line.

20. (Original) The method of claim 17 wherein said scheduling history is determined by a general prediction scheme.

21. (Currently Amended) A set of instructions residing in a computer readable storage medium, said set of instructions capable of being executed by a processor to implement a method of processing a computer instruction in a replay loop comprising:
receiving an instruction output from an instruction queue;
checking a delay unit for a scheduling history for said instruction;
determining a wait time for said instruction;
placing said instruction in a delay queue, wherein said delay queue includes at least one delay line with a fixed wait time; and
writing said instruction to said scheduler after said fixed wait time.

22. (Original) The set of instructions of claim 21 wherein determining a wait time for said instruction is based on said scheduling history stored in said delay unit.

23. (Original) The set of instructions of claim 22 wherein placing said instruction in a delay queue comprises:
matching said wait time for said instruction to the closest said fixed wait time for said at least one delay line.

24. (Original) The set of instructions of claim 21 wherein said scheduling history is determined by a general prediction scheme.